

# The Certainty of Uncertainty

by Mike Cohn • 0 Comments • originally published in Better Software on 2006-07-01

At the start of a new project there is an astounding amount of uncertainty. We're uncertain about how the technology we've selected will match the problem we're attempting to solve. If it's a new technology, we're uncertain about exactly how (and sometimes if) it will work. Even though the whiteboards in our shared work areas are covered with UML diagrams and other drawings, we're uncertain about how closely the design implemented in the code will match the design we've drawn on the walls. Most significantly, we're uncertain about exactly what it is we are developing.

In his book *Simultaneous Management*, Alexander Laufer groups uncertainties into two categories: means uncertainty and end uncertainty. Means uncertainty refers to our uncertainty about exactly how we'll design and develop a product; end uncertainty refers to uncertainty about exactly what a product will do. I have never seen a project team address a non-trivial problem and deliver at the end exactly what members thought they'd deliver when they began. There are always some amounts of what I call *emergent requirements*—those requirements that users don't realize they have until they begin seeing preliminary, incremental deliveries of a system. Upfront thinking cannot identify emergent requirements—and every project has some.

Uncertainty is guaranteed to exist at the start of a new project. How a team approaches uncertainty tells me a great deal about whether they are likely to successfully deliver a product users will want. One common approach to removing uncertainty is to focus initially on removing all end uncertainty (about what will be developed), and once that has been completely eliminated shift to eliminating means uncertainty. The team that wishes to eliminate all end uncertainty usually does so with a lot of upfront thinking about what users are likely to want. Some teams may do this through good practices such as user interviews and extensive user experience design.

“Every non-trivial system has some degree of emergent requirements. And so, it is impossible to eliminate end uncertainty through upfront thinking.”

However, regardless of how good the practices are that the team uses in this upfront attempt to

remove end uncertainty, the team is failing to acknowledge that every non-trivial system has some degree of emergent requirements. And so, it is impossible to eliminate end uncertainty through upfront thinking.

Attempting to remove all end uncertainty before beginning the true development work of a project is a classic mistake of teams using a waterfall process. Teams that choose to work iteratively (including but not limited to agile teams) attempt to concurrently reduce both end and means uncertainty. These teams understand that it is impossible at the start of a project to eliminate all uncertainty about what the product is to be. Parts of the product need to be developed and shown to customers, feedback needs to be collected, opinions refined, and plans adjusted. This takes time. While this is occurring, the team will be learning more about how it will develop the system.

The best way to deal with uncertainty is to iterate. To reduce uncertainty about what the product should be, work in short iterations and show (or, ideally give) working software to users every few weeks. Uncertainty about how to develop the product is similarly reduced by iterating. For example, missing tasks can be added to plans, inadequate designs can be corrected sooner rather than later, bad estimates can be amended, and so on.

Pause and reflect on your current project for a moment. Have you taken an appropriate approach to reducing both means and end uncertainty? Or have you fallen into the trap of thinking either type of uncertainty can be removed before the code is written? One thing you can be certain of is that an iterative approach to reducing uncertainty is the right approach.

---

Posted: July 1, 2006

**Tagged:** sprinting, sprint planning, management, customers

---

